

算法的优化:基于适应度方差的涡流搜索方法^①

刘魏岩^{②*} 邹立颖^{*} 王红艳^{**}

(^{*} 齐齐哈尔大学通信与电子工程学院 齐齐哈尔 161006)

(^{**} 齐齐哈尔大学机电工程学院 齐齐哈尔 161006)

摘 要 研究了算法的优化方法。为克服涡流搜索(VS)方法在优化运行时可能会陷入局部最优的问题,提出了一种基于适应度方差(FV)的涡流搜索(VS)方法,简称为 FVS 算法。该方法每次迭代时计算一次全体备选解适应度值的方差大小,如果适应度方差小于某一设定阈值,则调整涡流搜索方法备选解产生的位置,使方法产生的备选解能够跳出局部最优的限制。将这种基于适应度方差的涡流搜索方法与原始涡流搜索方法和其他几种优化算法对 8 个测试函数进行了寻优对比,结果验证了这种 FVS 方法的有效性。

关键词 算法优化, 涡流搜索(VS)算法, 适应度, 方差

0 引 言

算法(algorithm)是指解题方案的准确而完整的描述,它代表解决问题的策略机制。算法优化是指对算法的有关性能如时间复杂度、空间复杂度及正确性、鲁棒性等进行优化,使其具有更好的泛化能力,常见的有群智能优化算法。群智能优化算法具有原理易于理解、鲁棒性较强和优化性较高^[1]的特点,在图像检索^[2]、物流仓储^[3]、自动控制^[4]等科学领域得到了广泛的应用。近年来新型的优化算法也在不断提出,如反射优化算法^[5]、电磁场优化算法^[6]和狮群优化算法^[7]等。涡流搜索(vortex search, VS)算法^[8]是由 Doğan 和 Ölmez 提出的一种优化算法,它通过模拟水中波纹而对解空间进行搜索。涡流搜索算法具有参数少、迭代快、寻优效果较好等特点。但与其他优化算法类似,该算法运行时存在陷入局部最优的可能。涡流搜索优化算法产生的备选解随着迭代的进行会逐渐聚拢到当前迭代圆心周围,这样在算法运行后期则主要进行局部搜

索。较小范围的局部搜索易受迭代圆心控制而陷入局部最优。为了增强算法的局部寻优能力,跳出局部最优限制,可以在算法运行时适当改变当前迭代圆心的位置,来改善算法的寻优效果。

研究表明,通过统计全部备选解的适应度方差可以判断算法是否陷入局部最优^[9,10],本文利用这一原理提出了一种基于适应度方差的涡流搜索(fitness variance-vortex search, FVS)的优化算法,即 FVS 算法。该算法统计群体适应度值的方差,若适应度方差小于设定阈值则说明算法可能陷入局部最优,需要调整迭代备选解的产生位置。为测试改进后的涡流搜索优化算法的寻优性能,使用改进算法对 8 个测试函数进行寻优,并与原始算法和其他三种优化算法进行了对比。寻优结果验证了这种改进的涡流搜索优化算法的有效性。

1 涡流搜索优化算法

涡流搜索优化算法对解空间进行搜索前先利用公式

① 黑龙江省留学归国人员科学基金(LC2017028)资助项目。
② 男,1989 年生,硕士;研究方向:优化算法,电工电子技术等;联系人,E-mail: l_weiyang@163.com
(收稿日期:2017-10-09)

$$\mu_0 = \frac{e_{\text{lower}} + e_{\text{upper}}}{2} \quad (1)$$

计算解空间中心点 μ_0 , 其中 $e_{\text{lower}} \setminus e_{\text{upper}}$ 分别为解空间的下、上界。

$$p(x | \mu, \Sigma) = \frac{1}{\sqrt{(2\pi)^d |\Sigma|}} \exp\left\{-\frac{1}{2}(x - \mu)^T \Sigma^{-1}(x - \mu)\right\} \quad (2)$$

的随机备选解, 其中 $\Sigma = \sigma^2 \cdot [I]_{d \times d}$ (d 为解空间维数)、 $\mu = \mu_0$, 则 μ_0 为产生初始备选解的圆心。

σ 初始值 σ_0 由下式确定:

$$\sigma_0 = \frac{\max(e_{\text{upper}}) - \min(e_{\text{lower}})}{2} \quad (3)$$

其中 $e_{\text{lower}} \setminus e_{\text{upper}}$ 分别为解空间第 i 维的下界和上界。

初始解的搜索半径 $r_0 = \sigma_0 \cdot (1/x) \cdot \text{gammaincinv}(x, a_0)$ (其中 $x = 0.1, a_0 = 1$)。在初始化后每次迭代过程中搜索半径 r_i 按公式

$$r_i = \sigma_0 \cdot (1/x) \cdot \text{gammaincinv}(x, a_i) \quad (4)$$

自适应减小, 其中 $a_i = a_0 - \frac{i}{\text{MaxItr}}$, MaxItr 为最大迭代次数。

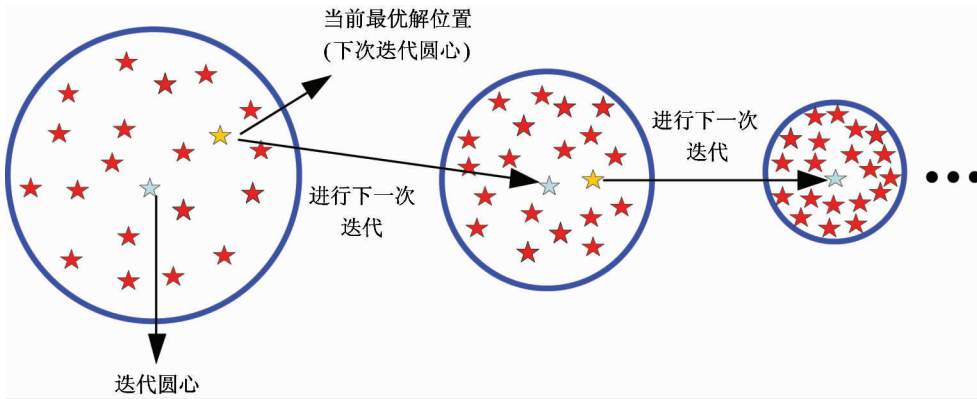


图2 涡流搜索优化算法原理示意图

2 基于适应度方差的涡流搜索算法

涡流搜索算法产生的备选解虽然具有随机性, 但在算法迭代时圆心 μ_i 可能会陷入局部最优。统计全体备选解的适应度方差 $\sigma_{\text{Fitness}}^2$ 并设定一个阈值, 当 $\sigma_{\text{Fitness}}^2$ 小于该阈值时, 则以解空间内 μ_i 的反向

$(1/0.1) \cdot \text{gammaincinv}(0.1, a_i)$ 与迭代次数 i 的关系如图1所示, 观察图1可发现由式(4)确定的 r_i 会随着迭代的进行而逐渐减小, 实现自适应调整。

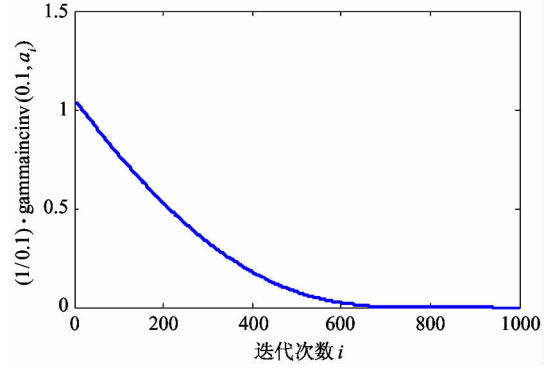


图1 gammaincinv 函数关于参数 a 的曲线特性

算法开始运行后, 将每次迭代最优备选解 best_i 与当前最优备选解 gbest 的适应度值大小作比较。选择 best_i 与 gbest 中适应度值小更新 gbest , 并且以 gbest 作为迭代的圆心 μ_i 。并以 μ_i 为圆心、自适应减小的 r_i 为半径产生新的满足高斯分布的备选解, 选出适应度值最小的备选解更新 μ_i 位置, 如此循环直到算法结束。输出最优解, 如图2所示。

位置作为下次迭代搜索的圆心, 使算法备选解位置能够跳出局部最优, 增强算法的搜索能力。

同时也应当注意图1和式(4)中迭代次数 i 与搜索半径 r_i 的关系。算法迭代后期, r_i 受 $(1/x) \cdot \text{gammaincinv}(x, a_i)$ 影响而取值变小。说明此时备选解被限定在以 μ_i 为圆心、 r_i 为半径的较小范围内, 主要进行局部寻优。利用适应度方差调整 μ_i 位

置也应在此时发挥作用,使备选解位置能够跳出局部最优。而在算法前期,应减弱这种作用以确保算法的全局搜索效果,所以需要用迭代次数来限制 μ_i 位置的选取。

具体方法如下。

设第 k 个备选解的适应度值为 f_k , 全体 f_k 的均值为 $\bar{f}$, 则适应度方差为

$$\sigma^2_{\text{Fitness}} = \sum_{i=1}^N \left[\frac{(f_k - \bar{f})}{f} \right]^2 \tag{5}$$

其中 f 为归一化因子,取值为

$$f = \begin{cases} \max\{|f_k - \bar{f}|\}, & \max\{|f_k - \bar{f}|\} \geq 1 \\ 1, & \text{其他} \end{cases}$$

设解空间有 d 维,当前迭代最优解位置为 best_i , 算法历次迭代搜索到的最优解位置为 $g\text{best}$, 当 $\sigma^2_{\text{Fitness}}$ 小于设定阈值时,令

$$\mu_i = (2 \times \mu_0 - \text{best}_i) \tag{6}$$

其中 μ_0 为解空间中心位置。即以解空间 best_i 的反向位置作为下次迭代的圆心 μ_i 。

以下式限制迭代次数与 μ_i 的位置

$$\mu_i = (1 - a_i) \times \mu_i + a_i \times g\text{best} \tag{7}$$

其中 a_i 取值与式(4)中相同。随着算法运行, a_i 取值由 1 到 0。式(6)是当 $\sigma^2_{\text{Fitness}}$ 小于设定阈值时,计算 μ_i 的第一步,式(7)是计算 μ_i 的第二步。式(7)中算法前期 $g\text{best}$ 对 μ_i 位置起主要作用;算法后期

best_i 的反向位置对 μ_i 位置起主要作用。这样就实现了用迭代次数来限制 μ_i 位置的选取。

基于适应度方差的涡流搜索优化算法流程如下。

步骤 1 算法初始化,计算初始圆心位置 μ_0 , 并在 d 维搜索空间产生 n 个备选解。

步骤 2 找出 n 个备选解中适应度值最小的为 μ_i 。

步骤 3 按式(4)计算搜索半径 r_i 。

步骤 4 用式(5)计算 $\sigma^2_{\text{Fitness}}$, 若 $\sigma^2_{\text{Fitness}}$ 小于设定阈值,则按式(6)和(7)调整 μ_i 位置。

步骤 5 以 μ_i 为迭代圆心,以 r_i 为搜索半径产生新的备选解。

步骤 6 判断算法是否满足停止条件,否则返回步骤 2。

步骤 7 输出最优解,算法结束。

3 仿真测试

为验证本文提出的改进涡流搜索优化算法的寻优效果,选择 8 个函数作为算法的测试函数(表 1)并与原始涡流搜索优化算法以及其他优化算法(粒子群优化(PSO)、萤火虫算法(FA)和人工蜂群(ABC)算法)进行对比。

表 1 8 个测试函数

函数	维数	搜索范围	理论值
$f_1(x) = \sum_{i=1}^n (x_i + 0.5)^2$	30	$[-100, 100]$	0
$f_2(x) = \sum_{i=1}^n (x_i)^2$	30	$[-100, 100]$	0
$f_3(x) = \sum_{i=1}^n (ix_i)^2$	30	$[-10, 10]$	0
$f_4(x) = \sum_{i=1}^n (ix_i)^4 + \text{random}[0,1)$	30	$[-1.28, 128]$	0
$f_5(x) = \sum_{i=1}^n x_i + \prod_{i=1}^n x_i $	30	$[-10, 10]$	0
$f_6(x) = \sum_{i=1}^n (\sum_{j=1}^i x_j)^2$	30	$[-10, 10]$	0
$f_7(x) = \frac{1}{4000} \sum_{i=1}^n x_i^2 - \prod_{i=1}^n \cos(\frac{x_i}{i}) + 1$	30	$[-600, 600]$	0
$f_8(x) = -20 \exp(-0.2 \sqrt{\frac{1}{n} \sum_{i=1}^n x_i^2}) - \exp(\frac{1}{n} \sum_{i=1}^n \cos(2\pi x_i)) + 20 + e$	30	$[-32, 32]$	0

仿真软件使用 Matlab 2012a,为统一测试条件,设算法迭代 2000 次,备选解数量 40 个,其中 FVS 算法 $\sigma_{\text{Fitness}} = 1 \times 10^{-3}$, PSO 算法的惯性权重 $\omega = 0.9$,

$c_1 = c_2 = 2$; FA 算法 $\alpha = 0.5, \beta = 0.2, \gamma = 1$ 。这 5 种算法均运行 30 次,并在表 2 中记录各算法寻优的平均值、最优值、最差值、标准差和运行时间。

表 2 5 种优化算法寻优结果对比

函数	算法	平均值	最优值	最差值	标准差	运行时间(s)
f_1	FVS	0	0	0	0	0.4246
	VS	3.6667	1	10	2.1867	0.2784
	PSO	2.7333	0	6	1.6595	3.8005
	ABC	0	0	0	0	15.4061
	FA	0	0	0	0	11.4851
f_2	FVS	3.5436e-32	1.2339e-32	8.2104e-32	1.7663e-32	0.3424
	VS	6.5373e-13	2.0503e-14	3.8654e-12	8.5880e-13	0.2204
	PSO	0.0800	0.0347	0.1320	0.0238	3.6237
	ABC	6.1803e-16	3.1965e-16	9.4317e-16	1.3026e-16	14.9526
	FA	0.0012	7.7810e-04	0.0019	2.4778e-04	14.2246
f_3	FVS	2.0176e-32	5.4367e-33	5.2072e-32	1.2530e-32	0.3624
	VS	2.2939e-04	75365e-08	0.0019	4.6647e-04	0.2267
	PSO	2.7829	0.9735	6.2836	1.3919	3.7018
	ABC	6.4266e-16	4.5914e-16	7.5754e-16	9.1304e-17	15.0675
	FA	0.0078	2.8623e-04	0.0755	0.0142	14.4761
f_4	FVS	0.0012	4.4414e-05	0.0043	0.0011	1.5448
	VS	0.0254	0.0072	0.0490	0.0114	0.7944
	PSO	0.2713	0.1197	0.5312	0.1038	4.1311
	ABC	0.1611	0.0863	0.2506	0.0364	21.5639
	FA	0.0319	0.0042	0.1007	0.0224	14.7902
f_5	FVS	1.4376e-16	7.5362e-17	2.7030e-16	4.0337e-17	0.3904
	VS	0.0015	3.3006e-06	0.0250	0.0050	0.2342
	PSO	1.8099	1.2200	2.5510	0.3909	4.2131
	ABC	2.2005e-14	7.1878e-15	5.2714e-14	9.6321e-15	15.5232
	FA	0.0486	0.0145	0.1492	0.0357	14.8668
f_6	FVS	4.1031e-30	2.2199e-30	6.4085e-30	1.1580e-30	1.1256
	VS	0.2250	0.0296	1.0307	0.2246	0.6126
	PSO	12.5025	6.3344	24.5031	4.2811	16.6014
	ABC	8.0284e + 03	2.3910e + 03	1.1944e + 04	1.8990e + 03	23.5630
	FA	40.4038	2.9891	174.8882	39.1643	27.2385
f_7	FVS	0	0	0	0	0.5784
	VS	0.0162	2.7562e-09	0.0416	0.0100	0.3457
	PSO	0.0261	0.0056	0.0674	0.0145	4.4983
	ABC	3.2858e-04	0	0.0099	0.0018	16.2573
	FA	0.0031	0.0015	0.0101	0.0019	15.1066
f_8	FVS	8.8818e-16	8.8818e-16	8.8818e-16	0	0.5837
	VS	0.0832	2.2047e-08	1.3404	0.3175	0.3470
	PSO	1.8379	0.4642	3.1489	0.7118	4.6094
	ABC	5.1793e-13	2.2116e-13	1.0596e-12	1.8750e-13	16.3640
	FA	0.0085	0.0064	0.0116	0.0013	15.3556

通过表 2 可以看出,FVS 优化算法对 8 个测试函数 30 次的寻优均值和标准差都要小于原始 VS 优化算法和其他对比算法,这说明 FVS 优化算法在寻优精度和稳定性上都有一定提升。按式(6)和(7)的设定,在 FVS 算法运行前期,圆心位置 μ_i 的选取主要依赖于历次迭代搜索的最优位置 g_{best} 。此时迭代圆心 μ_i 的选取与原算法相近,全局寻优能力较强。随着算法不断迭代, a_i 的值由 1 减小到 0,此时圆心位置 μ_i 的选取受 μ_i 的反向位置影响增强。在进行局部寻优时备选解的搜索范围更广,能够跳出局部最优的限制,使算法在相同迭代次数下更易找到最优解。但是由于 FVS 算法加入了计算适应度方差和迭代圆心 μ_i 调整等步骤,使运行时间相较于 VS 算法有所增加,但与其他对比算法相比依然保持了 VS 算法的快速性。综上所述,基于适应度方差的涡流搜索优化算法具有较好的寻优精度和稳定性,能够使算法跳出局部最优限制,增强了算法的寻优性能。

4 结 论

本文结合备选解适应度方差与反向原理,在涡流搜索优化算法基础上提出了一种具有迭代圆心调整机制的涡流搜索优化算法。通过对 8 个测试函数进行寻优验证,结果表明基于适应度方差的涡流搜索优化算法相较于原始涡流搜索优化算法具有更好

的寻优性能。

参考文献

[1] 刘琼. 智能优化算法及其应用研究:[硕士学位论文] [D]. 无锡:江南大学物联网工程学院, 2011

[2] 周恒俊. 智能优化算法及其在图像检索中的应用研究:[硕士学位论文] [D]. 济南:山东大学信息科学与工程学院, 2016

[3] 史朋涛. 基于智能优化算法的人工拣货仓库的仓储问题研究:[硕士学位论文] [D]. 长春:长春工业大学工业工程系, 2017

[4] 李恒. 改进的萤火虫算法及其在 PID 控制器参数整定中的应用:[硕士学位论文] [D]. 合肥:安徽大学计算机科学与技术学院, 2017

[5] Kashan A H. An effective algorithm for constrained optimization based on optics inspired optimization (OIO) ☆ [J]. *Computer-Aided Design*, 2015, 63:52-71

[6] Abedinpourshotorban H, Shamsuddin S M, Beheshti Z, et al. Electromagnetic field optimization: a physics-inspired metaheuristic optimization algorithm[J]. *Swarm & Evolutionary Computation*, 2016, 26:8-22

[7] Yazdani M, Jolai F. Lion optimization algorithm (LOA): A nature-inspired metaheuristic algorithm[J]. *Journal of Computational Design & Engineering*, 2016, 3(1):24-36

[8] Doğan B, Ölmez T. A new metaheuristic for numerical function optimization: vortex search algorithm[J]. *Information Sciences*, 2015, 293: 125-145

[9] 周燕, 刘培玉, 赵静, 等. 基于自适应惯性权重的混沌粒子群算法[J]. 山东大学学报(理学版), 2012, 47(3):27-32

[10] 虎涛涛, 单要楠. 一种混沌变参数粒子群优化算法[J]. 自动化仪表, 2017, 38(3):37-40

A fitness variance based vortex search method for algorithm optimization

Liu Weiyan *, Zou Liying *, Wang Hongyan **

(* College of Telecommunication and Electronic Engineering, Qiqihar University, Qiqihar 161006)

(** College of Mechatronic Engineering, Qiqihar University, Qiqihar 161006)

Abstract

The algorithm optimization is studied. In order to overcome the problem that the vortex search (VS) method for algorithm optimization may fall into local optimum in operation, a new vortex search method based on fitness variance, called the FVS, is proposed. The FVS method calculates the variance of fitness value of all alternative solutions at each iteration. If the calculation result is less than a set threshold, the position of the alternative solution of the vortex search method is adjusted. The alternative solution generated by the method can jump out of the limit of the local optimal. This optimization method is compared with the original vortex search method and other methods on 8 test functions, and the results verify the effectiveness of the proposed method.

Key words: algorithm optimization, vortex search (VS) algorithm, fitness, variance